
CHINESE CHARACTER PROCESSING SYSTEM BASED ON CHARACTER-ROOT COMBINATION AND GRAPHIC PROCESSING

FAN CUN-CHANG*, PAOLO ZINI**

* *Shanghai Institute of Ceramics, Chinese Academy of Science, Shanghai, China*

** *CNUCE, Institute of CNR, Pisa, Italy*

ABSTRACT

This paper deals with the problem of Chinese character processing. An introductory description is given. The existing solutions are discussed and their limitations pointed out. A new system for Chinese character processing based on character-root combination and graphic processing is proposed. A prototype implementation based on a microcomputer system is described. The memory size occupied is small and can fit even in a personal computer. The input rules are simple and can be easily accepted by untrained users. With 330 Chinese character-roots and a few combination rules the system can produce more than 8,000 Chinese characters

1. Introduction

The first consideration to be made before starting on a project of this kind is the difference in composition of a Chinese character (in the sense of a word) and that of a word in any Western language. In the latter case, all words are formed by the composition of alphabets taken from a list of a few dozen. Chinese characters can, at most, be split up into minor radicals generally known as character-roots. From the phonetics point of view, the pronunciation of a word in Western languages depends largely on the characters with which the word is spelt; in Chinese, there is little dependence of the pronunciation of a character on the character-roots composing it. Moreover, it often happens that a number of quite different characters may have the same pronunciation.

The structure of Chinese characters is very complex. Character-roots have variable positions in different characters. Different compositions of the same character-roots may produce distinct characters. So the problem of writing Chinese characters with a computing system is quite complex.

The "Kang Xi Dictionary" edited in 1716 contains more than 47,000 characters. The authoritative "Chinese Dictionary" edited in 1915 includes more than

48,000 characters. At present, the number of commonly used Chinese characters is about 5,000 to 8,000 [1, 2, 3].

Chinese character information processing is becoming an important subject in the field of computer science, and in the last ten years, an ever increasing number of computer scientists are involved in this subject.

1.1 *State of the art*

Research has been done on many methods to generate Chinese characters on the computer [4–8]. A commonly used method is to have a Chinese character library in memory, where the characters are stored in dot-matrix patterns. A special coding scheme is adopted to address the characters.

A great variety of coding schemes are in use.

- The "Standard code" is based on the standard telegraph code of Chinese characters used in China for telegraph transmission [9]. Each character is coded with four decimal digits, and it is difficult for the ordinary person to be familiar with these "meaningless" codes.

- The "Big Keyboard" is derived from the big character case of Chinese character typewriters [4]. The code character in this case is the character itself. The big keyboard is compactly laid out in order to reduce its size, and the keys are labelled with the respective Chinese characters. By touching a key with a pencil-like probe the corresponding character is displayed.

The "Big Keyboard" is more easily accepted by users, as they can look up the big keyboard and, however slowly, finally find the character they want. The keys have their definite meanings. The drawback is that it is a tedious job for the average users to pick a character up from more than 2,000 characters. Besides, it is very difficult to produce characters which do not exist on the keyboard or in the memory.

- The "Pinyin Code" is based on the standard pronunciation of Chinese characters expressed by an alphabet similar to the English one. However, under the same Pinyin code there may be several Chinese characters. Other complementary actions must be taken to eliminate the ambiguity [7]. Normally the code of the radical or of the character-root may be added, then we obtain a scheme which combines pronunciation with form.

This method can reduce the number of keys to 300, or a standard keyboard may even be used, but the input rules are not simple. Besides, people who are not familiar with the Standard Chinese Pronunciation would find this method difficult to apply.

- Coding according to character form seems at this point more acceptable. The components of the characters, or the character-roots, are studied and employed in character processing system.

The "Double character-root code" Chinese character input system developed by the Shanghai Jiaotong University is a typical example. It is popular among users thanks to its audio-visual format and simple rules [10].

- The combination of character-root with pronunciation also seems adoptable. "Seeing Character is Knowing Code", proposed by Dr. Zhi Bing-yi, is a famous coding scheme of this type [10]. The code of the character is the combination of the first letter of the Pinyin code of the character-roots composing it.

The "Ink jet printer for Chinese characters" developed by the Olympia Company adopts this kind of code, and the input keyboard is just a standard one. Of course, the practical input rules are not simple.

In general, all these methods use a character library and a large size of memory would be needed. With 4,000 Chinese characters in 32x32 dot-matrix pattern, the library would occupy about 500k bytes. This is a disadvantage which should not be ignored.

Graphic processing is another way to realize the Chinese character processing [6]. However, in the past the required hardware and software were considered comparatively complex and only few people were interested in it.

2. Our Approach

We believe that we must first consider the needs and characteristics of the potential users of a Chinese writing system before making any attempt to find a solution to the Chinese writing problem. After this evaluation we can try to analyze the problem by following as much as possible the "natural" way. Practically we believe that the application must work according to man needs and not viceversa. We have analyzed the method of Chinese writing used in Chinese schools to deduce the "mental" approach of Chinese people to writing and tried to design our Chinese writing system according to that; the results obtained are promising.

2.1 The requirements

There is a high number of potential users of this system in offices, in mass media, in printing, in libraries and in all jobs where information retrieval is requested. In these environments, complex machines requiring trained and skilled users are difficult to use and are often not accepted.

From the hardware point of view, most of the above mentioned applications

need writing systems, terminals for medium or large size computers, and above all, personal computers.

On the basis of these considerations and after an examination of the existing solutions and their limitations, we deduce some basic requirements.

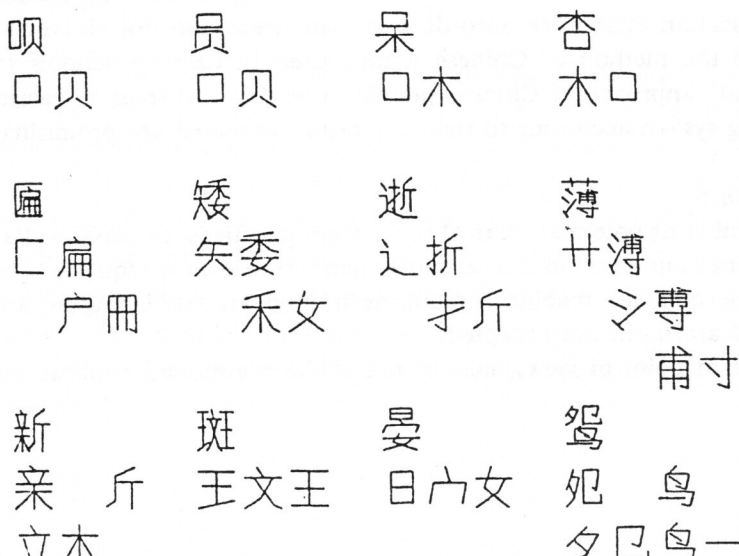
- The method, which has to be used on equipment of mass production, must be very cheap;
- the keyboard must be of a simple and compact structure. Keyboards with over 200 physical keys are expensive and require long training periods.
- the system should be able to produce at least the 8,000 characters which form the basic Chinese vocabulary, and should, moreover, offer possibilities of future expansion;
- the coding system should be simple and intuitive so that a "normal" user with no particular training does not encounter great difficulties.

2.2 "School" method on Chinese writing

Let us now consider the approach commonly adopted in most Chinese schools to teach children Chinese writing, and point out some interesting aspects of this method.

Each character is contained graphically in a rectangular of a fixed dimension and is considered as a single unit. However, every character is made up of one or more components, which may be defined as character-roots, each occupying a certain position in the rectangular according to a series of combination rules. The character-roots forming a character may acquire another meaning in another character if their position is changed (see Fig. 1a). It can be noticed that a character formed by some "simple" character-roots may appear as a "complex" char-

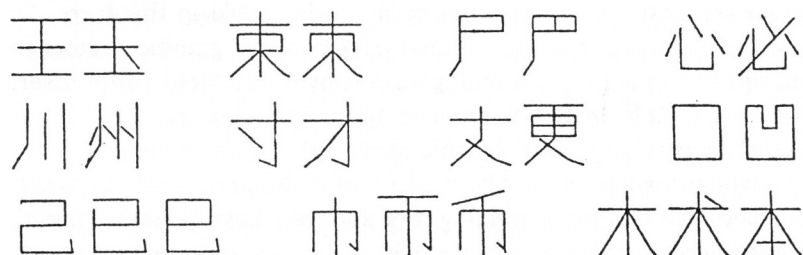
Figure 1



acter-root in another character. This concept of "complex" character-roots is exploited in our system to facilitate the combination process (see Fig. 1b). A more detailed explanation will be given later.

A closer study reveals that some character-roots are graphically modified versions of other character-roots. This is done by adding or cancelling one part or one stroke of the "mother" character-root(s) (see Fig. 2).

Figure 2



At this point, we can say that by a repetitive composition process of the character-roots, different layers of complexity in root construction are possible; i.e., we may obtain a more or less complex character-root with a greater or smaller number of simple character-roots. We have preferred to take the school level as our standard, as this is the most representative one for the ordinary person.

At this level, the combination possibilities which are to be considered as combination rules in our system are few: the roots may appear side by side, or one on top of the other, or one root may be "enveloped" in another. In these combination procedures, the roots may be compressed or they may change their dimensions according to the case, while maintaining their graphic structure.

2.3 The algorithm

On the basis of the previous analysis, we have developed an algorithm which uses a table of character-roots, some composition operators and some modifying operators, or simply modifiers. A table of 330 roots has been obtained from the Chinese Dictionary. If we attempt to derive a specific combination rule for each type (the graphic arrangement) of character, we would have too many rules for any practical application. We have chosen instead the idea of repeated applications of composition operators, storing the intermediate result of the composition in a temporary memory. This "temporary root" can be used again in further compositions with other simple or "complex" roots. Only the final result will appear on the screen and the composition operators behave practically like arithmetic operators. The composition operators adopted are those described in the previous section, namely side by side combination, stacking and enveloping.

In order to simplify the table of character-roots and to adhere as much as possible to the school standard, we have provided a series of modifiers which, as it has been mentioned above, are used to add, cancel or change a stroke or a part of one character-root to form another (see Fig. 2). These modifiers not only reduce the size of the table, but also solve some graphic problems whose solution would otherwise be impossible. To expand the capability of the algorithm other operators or modifiers can be added, each operator requiring only a few more statements for its implementation.

The character-roots cannot be stored in dot matrix form because they are subject to complex dimensional changes not easily performable in this form. So we have preferred to store and manipulate character-roots as graphic vectors so that a change in dimension is simply a multiplication by a size factor. Moreover, the small size of roots table makes it possible to use multiple roots tables to change the "character font", i.e., the graphic aspect of the characters.

The resulting keyboard will generate about 350 root codes and a few operators. The keys are labelled with the corresponding root and they have a more "direct" impact on the user, since they are a part of a Chinese character, and not simply "meaningless" codes.

The problem remains as to whether it is compulsory for a user to always insert the operators together with the root, or whether these operators could become implicit if the roots are typed in a certain sequence.

After a first analysis of the dictionary, we discovered that in most cases, certain character-roots in a given sequence can result in one and only one "meaningful" character. Only a very limited number of sequences of character-roots can produce different possibilities of characters, the so-called ambiguous cases, which can easily be singled out during program writing.

Probably the best compromise is that of using a parser for the common cases and specified operators for the ambiguous ones. However, the analyses done up to now are not sufficiently thorough for us to define a context free grammar for the method [12, 14].

3. Implementation

To verify the functionality of the method an experimental implementation has been carried out. The prototype uses a graphic terminal Tektronix 4013 and a microcomputer ONIX 8002 using the UNIX V7 operating system. The tools are not chosen for any particular reasons but simply because they are available at the center where the authors are working.

The first version of this prototype shows that the method is simple enough to be easily adapted to simpler microcomputers without any loss of efficiency. The display screen used is a graphic storage tube, but the same features can be found on several other cheaper personal computers. -

3.1 *The graphic representation*

As we have stated before, the roots change their dimensions relatively to the position and space they occupy in the final character. This is done with the graphic vector representation of the roots. The root is represented by a sequence of pairs of coordinates each of which specifies respectively the start and the end of every graphic segment of the root. The sequence is ended by a terminator symbol. Optional information is located after the terminator, to be used for some composition or modification operations, like the "envelope" composition, in which the information will be the size of the "box". If the root has modified versions the additional information will be the modification to be applied. The roots table holds 330 roots and its size is about 15 kbytes; an optimized organization of the table can reduce its size to less than 5 kbytes. The total size of the object code is about 40 kbytes.

3.2 *The keyboard*

The keyboard (a prototype has been implemented) has about 120 physical keys for the roots, a couple of keys for shifts, about 10 keys for composition or modification operators plus all the classical function keys of a keyboard (carriage return, line feed, backspace etc.). The roots have been encoded, in an absolutely arbitrary way, by using pairs of lowercase ASCII characters; the operators are encoded with other ASCII characters and space is used as a character terminator symbol. This means that a Chinese text is encoded as a sequence of ASCII characters and is fully compatible with all operating systems.

A lexical-analyzer [12, 13] program translates these pairs into roots table addresses and the symbol representing the operators into special codes. The output of the lexical-analyzer is the input of the program which implements the algorithm described in the next section.

3.3 *The algorithm implementation*

The algorithm is implemented by a program written in the C language [11]. It accepts as input strings of roots, composition operators, modifiers, and terminators. The operators can be unary or binary and are all postfixes, so that the input string looks like an equation expressed in reverse Polish notation. A stack is used to hold the roots and the "composed" roots resulting from previous composition operations. The program scans the input string and, when a root is found, its vector representation is pushed into the stack. When a unary operator is found, the last pushed root is popped out, the operation is performed on it and the resulting root is again pushed into the stack. Analogously, when a binary operator is found, the two last pushed roots are popped out, the operator applied and the resulting "complex" root pushed into the stack. When the terminator is found, the stack is scanned: if the stack contains only one root it is displayed by a standard graphic package, otherwise the roots contained in the stack are composed

according to the final composition rule and the resulting character is displayed. In our prototype, the final default operator is a "side by side" composition operator, and can compose up to three roots. In case of vertical composition; i.e. stacking, the final composition operator must be specified.

The basic operators are:

- side by side composition
- stacked composition
- enveloped composition
- modify by suppression of one stroke
- modify by addition of one stroke
- modify by replacement of one stroke with one or more different strokes.

To simplify the writing we have added one more operator for a special composition (Fig. 3). This operator is not actually necessary as the same result can be obtained by using two basic operators. However, it implements a usual construction of Chinese writing. The implementation of composition operators is not complex; each operator requires only a few statements, and each operator executes only compressions (multiplication by coefficient), shifts (sum of a constant), and merging of vectors of coordinates representing the roots into a single vector.

Figure 3



This implementation has proven to be rather powerful and relatively easy to use: all the 8,477 characters listed in The XINHUA Dictionary have been written out in different sizes and the needs of general Chinese character processing purposes are met.

3.4 Plans of future development.

We believe that it is necessary to first test the applicability of this method to small computers. Our schedule covers a second personal-computer based prototype. During its implementation, due consideration will be taken of the implementation cost of necessary graphic functions. As to the algorithm, the composition rules are very simple, but the elimination of the explicit specifications makes the system even simpler.

It is therefore our intention to continue our research on a parser capable of deducing composition rules from the sequences of the roots, without specifying the operators.

4. Requirements Matching

At its present stage, the prototype seems to be very interesting and easy to implement.

The present version of the prototype was completed after 5 months of work. Most of the time was spent on preparing the roots table. The keyboard implemented has a relatively limited number of keys and seems to be quite easy for the typical user and relatively cheap.

The method of composition, being very close to the method of hand writing, seems to be simple and linear to understand, and does not request any previous knowledge of meaningless codes or of the exact pronunciation of the characters.

The simplicity of the system itself provides possibilities of producing different sizes (by specifying a parameter) (see Appendix) and different types of graphic characters (by loading different character-root-fonts), an advantage not found in existing solutions, and a particularly important factor in editing. Our experiments have shown that by adding some more character-roots into the character-root table it is possible to produce almost all of the 48,000 Chinese characters for special users. The size of the object code will be still moderate.

The hardware complexity can be evaluated mainly by following an intuitive approach. A video screen needs, in order to be graphic, a RAM memory bit for every point of the screen. In our case, a resolution of 1024×512 points will suffice; i.e., 64 Kbyte. The remaining part of the hardware is equivalent to that of the corresponding alphanumeric screen. A comparison of this memory size with the 500 Kbyte ROM needed for a medium size character library will show that our method can be much cheaper and less complex than existing ones.

5. Conclusions

Obviously some limitations can be found in our work. For instance, the roots table was built only on the basis of direct experiences acquired in this particular field. There has been little linguistic information from experts of this sector, and the root table is not an authoritative one. However, we do think that this new attempt offers a series of interesting points and may prove to be a further step in the realization of an efficient Chinese Character Information Processing system.

Acknowledgement

We wish to express our most sincere thanks to the Italian National Research Council (CNR) and the Chinese Academy of Science for the support given to this project, and in particular, to Prof. Xue Zhi-Lin and Mr. Li Zong-Jie of the Shanghai Institute of Ceramics, Chinese Academy of Science, who have kindly sent the authors precious information not otherwise available in Italy, and to Dr. S. Trumpy, Director of CNUCE, Institute of CNR, for his constant support. We also wish to thank Mrs. Tao Pei-lin for the suggestions and comments she has given us during the preliminary phase of our work and for the long experimental work she has carried out during the functionality test of our system.

References

- [1] The Collection of Coded Chinese Characters for Information Communication;
The 4th Ministry of Machine-Building, China; May 1, 1981.
- [2] XINHUA Dictionary;
SHANGWU Publishing House, China; 1981.
- [3] Chinese English Dictionary;
SHANGWU Publishing House, China; 1980.
- [4] Introduction of Chinese Character Processing;
Technology Reference Room of Shanghai Computer Factory;
Computer Technology n. 4, 1982.
- [5] The Processing Abstracts of the First Academic Discussion on Chinese Character Information Processing System;
Chinese Instrument and Apparatus Society; June, 1981.
- [6] Lian Jing: The Design and Realization of Chinese Character System Based on Graphic Processing; The Research and Development on Computer,
Institute of Computing Technology, Chinese Academy of Science.
- [7] Geng Li-da: Function and Conclusion on Chinese Character Processing System Based on PINYIN; Institute of Computing Technology, Chinese Academy of Science.
- [8] Shen Jia-lin; Microcomputer System for Chinese Character Information Processing;
Instrument and Control n. 2, 1983.
- [9] Standard Telegraph Code; Ministry of Post and Telecommunication, China.
- [10] GUANGMING Daily; April 18, 1983.
- [11] B.W. Kernighan & D.M. Ritchie: The C Programming Language;
Prentice-Hall, N.J., 1978.
- [12] A.V. Aho & J.D. Ullman: Principles of Compiler Design;
Addison-Wesley Publishing Company; 1977.
- [13] M.E. Lesk & E. Schmidt: LEX: A Lexical Analyzer Generator;
Murray Hill, N.J.; 1980.
- [14] S.C Johnson: YACC: Yet Another Compiler-Compiler;
Murray Hill, N.J.; 1978.

